Bridge **402**, the audio device **401** is established **608**a as the ACLINK controller to the codec **405**. The decoded audio is sent to the codec **405**. As part of transferring control from the primary device to the audio device **401**, the storage location (e.g. CD-RW **403**) directory is traversed and a play list of audio files (e.g. MP3) is built **609**a in alphabetical sequence. Normal audio CD (CD_DA format) is also supported. In the case of CD_DA format, playback is by track number as with a regular audio CD. Next to be examined is a more detailed discussion related to upgrading the audio device's **401** firmware.

**[0062]** As discussed above and illustrated in Figure **6**a, the audio device **401** is initialized during the power up process. During initialization checks are made to determine if there is an upgrade for the audio device's **401** controller and DSP firmware. In one embodiment, the audio device's **401** firmware can be upgraded in three ways: an external upgrade ROM, through an I2C interface, and through a USB interface. Referring briefly back to Figure **4**, an embodiment of the architecture disclosed within a system is illustrated. Figure **7a** illustrates an alternate embodiment of the audio device of Fig. **4**. Referring briefly back to Figure **5**b, an embodiment of a method of upgrading the firmware of the audio device described is illustrated. An alternate embodiment of a method of upgrading controller and DSP firmware is illustrated in Figure **7b**.

**[0063]** Referring to Figures **7**a and **7**b, the audio device **700**a is powered on **701**b. Next, the controller (e.g. 8051) **701**a initializes **702**b system internal ROM **702**a. The controller firmware program then loads **703**b DSP boot-up code with standard functions from an internal ROM **702**b through the Controller to the DSP

gateway **703**a into DSP memories **715**a. The DSP standard functions include but are not limited to Playback MP3 bitstream and Playback audio PCM data. After loading the DSP bootup code, the controller **701**a checks **704**b an external ROM **718**a for a firmware upgrade. The external ROM **718**a is accessed through the audio device's **700**a master mode I2C port **705**a. If a valid upgrade is found in an external ROM **718**a, then the controller **701**a loads **705**b data from the external ROM **718**a in through the I2C port **705**a, and into an SRAM **707**a. In alternate embodiments the SRAM **707**a may be a register. In one embodiment the internal ROM **702**a which contains the DSP bootup code is not re-writeable.

**[0064]** After the new code is loaded to the SRAM **707**a, the controller **701**a checks **706**b its vector table to determine whether the DSP program to be loaded to DSP memory **715**a is located in the internal ROM **702**a, SRAM **707**a, or a combination of both. Next, the controller **701**a reads the upgrade program from the SRAM **707**a and loads **707**b the upgrade program through the controller Bus (e.g. 8051 Bus) **708**a and gateway **703**a to DSP program memory **715**a. Next, the DSP **709**a initializes **708**b itself and starts executing the program loaded into DSP program memory **715**a. The controller **701**a at this point enters **709**b into the main loop and monitors user-generated interrupts. While the audio device **700**a is in use, the audio device's **700**a firmware can be updated any time by a host processor in the system through the use of either the I2C port **706**a or USB port **710**a. During execution of the DSP program, when an interrupt request occurs, the controller **701**a begins to execute the interrupt service routine. After the interrupt has been serviced, the DSP **709**a continues program execution.

**[0065]** Referring to Figure **7**c, an embodiment of a method for upgrading the firmware of the embodiment described through an I2C port **706**a is illustrated. Referring to Figures **7**a and **7**c, the upgrade through the I2C port **706**a is initiated by the host sending **701**c an upgrade command and code through I2C port **706**a. The controller (e.g. 8051) **701**a then moves **702**c the 8051 firmware code into SRAM memory **707**a. If there is extended DSP code to be sent **703**c from the host it is sent through the I2C port **706**a. The controller **701**a moves **704**c the extended DSP code into DSP memory **715**a. After the 8051 firmware and DSP code is received, the host sends **705**c an execute new code command to start the new 8051 code execution.

**[0066]** Referring to Figure **7**d, an embodiment of a method for upgrading the firmware of the embodiment described, through the USB port **710**a is illustrated. Referring to Figures **7**a and **7**d, the upgrade through the USB port **710**a is initiated when the host sends **701**d an upgrade command and code through the USB port **710**a. The controller **701**a copies **702**d the 8051 firmware code into SRAM memory **707**a. If there is extended DSP code to be sent from the host it is be sent **703**d through the USB port **710**a. The controller **701**a copies **704**d the extended DSP code into DSP memory **715**a. After the 8051 firmware and DSP code is received, the host sends **705**d an execute new code command to start the new 8051 code execution.

**[0067]** Referring briefly back to Figure **5c**, an embodiment of a method of playing an audio file utilizing the audio file described is illustrated. Referring to Figure **8**a, an alternate embodiment of a method for playing an audio file from a

primary device's disk drive utilizing the embodiment described is illustrated. In an alternate embodiment the audio device may play an audio file from a hard drive. Referring to Figures **7**a and **8**a, to initiate the playing of audio files the play button on the audio device's **700**a attached keypad **704**a is pressed **801**a. Next, the audio device's **700**a internal ROM **702**a is checked **802**a to determine the function of the keypad entry received at the audio device's **700**a I/O port **711**a.

**[0068]** In one embodiment, the audio device **700**a has eight pins dedicated for a push-button keypad. These eight inputs are directly connected to an I/O port **711**a of the built-in controller (e.g. 8051) **701**a. In addition, the eight pins are OR'ed together to generate an interrupt signal. If any keypad entry is pressed, the signal goes low (active) and an interrupt is generated. Referring to Figure **8**b, a table of the functions associated with each keypad entry is illustrated. The audio device's **700**a internal ROM **702**a defines the functionality of the eight keypads. According to Figure **8**b, key2 **801**b is associated with the play button on the keypad **404**. The function of key2 **801**b is to toggle between the playing and pausing of an audio file.

**[0069]** After the controller **701**a determines that the play function has been chosen, the controller **701**a through the IDE interface **712**a sets up **803**a data transfer from the disk drive (e.g. CD-RW) **403** and determines the format (e.g. MP3) of the data to be transferred. The controller **701**a instructs **804**a the audio device's **700**a DSP **709**a to begin processing the data and provides the DSP **709**a with the data's format. The DSP **709**a determines **805**a if the data requires decoding (e.g. decompress MP3 file). If the data needs to be decoded, the DSP

709a begins pulling 806a the file data from memory. The DSP 709a then decodes 807a the file based on format information provided by the controller 701a. Whether the data requires decoding (e.g. MP3 file) or not (e.g. CD_DA format), after the DSP 709a has finished processing the data, it sends 808a the audio output through an I2S or AC-link 713a to a codec 714a. Next, the signal is output 809a as music through a listening device (e.g. computer's speaker).

**[0070]** Referring back to Figure **4**, an audio device **401** within a computer system is provided. Figure **8**c provides one embodiment of the audio device **401** of Figure **4** within a computer (e.g. notebook) audio system. The audio device **803**c is coupled to the computer's South Bridge **805**c, a CD-ROM **804**c, and a CODEC **801**c. Referring to Figure **8**c, the codec's **801**c digital to analog converter (DAC) converts the digital audio signal to an analog signal. The analog signal is then forwarded to an amplifier **802**c where the strength of the analog signal is boosted. After the signal has been amplified it is sent to a listening device (e.g. speaker).

**[0071]** In an embodiment of the architecture described, recording voice received through a microphone attached to the computer is possible. Referring briefly back to Figure **5**d, an embodiment of a method of recording voice utilizing the audio device described is illustrated. Referring to Figure **9**, an alternate embodiment of a method of recording voice utilizing the embodiment described is illustrated. Referring to Figures **7**a and **9**, the record function is chosen **901** from a keypad attached to the audio device **700**a. The audio device **700**a receives **902** the keypad entry at an I/O port **711**a and forwards the keypad entry to the

audio device's controller **701**a. After the controller **701**a determines that the keypad entry chosen is the record function, the controller **701**a sets up **903** data transfer from the codec (e.g. AC97) **714**a through the audio device **700**a to a disk drive (e.g. CD-RW) attached to the audio device **700**a through an IDE interface **712**a. In an alternate embodiment, a hard drive or a SmartMedia is used in place of a disk drive. The SmartMedia interface **716**a may be used to communicate with a SmartMedia. Also, the SD flash controller **717**a may be used to communicate with an SD flash memory. The voice to be recorded enters **904** the audio device **700**a from the codec **714**a at an I2S port or AC-link **713**a. The voice is then processed **905** by the audio device's **700**a DSP **709**a and passed through the controller to DSP gateway **703**a to the audio device's controller **701**a. The controller **701**a sends **906** the voice through an IDE interface **712**a to an attached disk drive. In alternate embodiments, the controller **701a** sends the voice through an interface to an attached hard drive.

**[0072]** In addition to recording voice the audio device also has in one embodiment, a karaoke feature. This feature provides for the playing of music utilizing the audio device's play feature described in Figures **5**c and **8**a while voice received through a microphone attached to the primary device is output at the same time. Referring briefly back to Figure **5**e, an embodiment of a method of utilizing the karaoke feature of the audio device described is illustrated. Referring to Figure **10**, an alternate embodiment of a method of utilizing the karaoke feature of the embodiment described is illustrated.

**[0073]** Referring to Figures **7**a and **10**, voice is received **1001** into a computer's microphone at the same time that the audio device's play function is being utilized. The voice is received **1002** into the computer's Codec (e.g.AC97) **714**a. In step **1003** a check to determine if the audio device is currently playing an audio file is made. In step **1004**, if the audio device is not currently playing an audio file, voice received through the microphone is processed by the CODEC and output through the speaker. In step **1005**, voice received through the microphone is processed by the Codec **714**a, mixed with the audio file, and output through a listening device (e.g. computer speaker) at the same time as the audio file being played is being output through the attached computer.

**[0074]** In the foregoing detailed description, the method and apparatus of the present invention has been described with reference to specific exemplary embodiments thereof. It will, however, be evident that various modifications and changes may be made thereto without departing from the broader spirit and scope of the present invention. In particular, the separate blocks of the various block diagrams represent functional blocks of methods or apparatuses and are not necessarily indicative of physical or logical separations or of an order of operation inherent in the spirit and scope of the present invention. For example, the various blocks of Figure 4 may be integrated into components, or may be subdivided into components. Moreover, the blocks of Figures 5a, 5b, 5c, 5d, 5e, 6a, 7b, 7c, 7d, 8a, 9, and 10 represent portions of methods which, in some embodiments, may be reordered or may be organized in parallel rather than in a

linear or step-wise fashion. The present specification and figures are accordingly to be regarded as illustrative rather than restrictive.